MPT USER MANUAL

Andrea Boccardi

June 6, 2005

Abstract

This is a simple manual for the use of the MPT test system. It describes the MPT board functionalities and registers from the user point of view. You will not find inside architectural desription of the board or of the FPGA design.

Contents

1 Introduction **6**

2 MPT registers **8**
2.1 STATUS register 8
2.2 COMMAND register 8
2.2.1 READ MEMORY command 8
2.2.2 START JTAG command 9
2.2.3 READ ADC command 10
2.2.4 PROGRAM DAC command 10
2.2.5 START DAQ command 11
2.2.6 READ REGISTER command 12
2.2.7 INITIALIZE MEMORY command 12
2.2.8 SYNCHRONIZE command 12
2.2.9 VALIDATE JTAG CHAIN command 13
2.2.10 ANAPIL SCAN command 13
2.2.11 DIGITAL PILOT TEST command 14
2.2.12 INVERT COMMAND POLARITY command 14
2.2.13 ANAPIL ADC SCAN ON command 14
2.2.14 ANAPIL ADC SCAN OFF command 14
2.2.15 USB mode 1 command 15
2.2.16 USB mode 2 command 15
2.2.17 JTAG internal interface command 15
2.2.18 JTAG external interface command 15
2.3 TDI and TDO registers 15
2.4 DAQ COMMAND register 15
2.5 JTAG VALIDATION LOG register 16
2.6 ANAPIL STATISTICS 16
2.7 ANAPIL STEP VALUE 16
2.8 ANAPIL STEP'S NUMBER 16
2.9 ANAPIL WAIT AFTER DAC 16
2.10 ANAPIL WAIT BETWEEN ADC 16

2.11 ANAPIL MCM VDD 16
2.12 JTAG mem 17

3 ANAPIL's DACs and ADCs scan and characterization 18

4 DIGITAL PILOT automated test 20

5 The VME interface to the MPT 23

6 The USB interface to the MPT 24

List of Tables

2.1 MPT's Registers 9
2.2 STATUS register's bit definition 10
2.3 COMMAND register's bit definition 10
2.4 Availeable commands 11
2.5 DAQ sequence number 0: synch sequence 11
2.6 DAQ sequence number 1: reset sequence 12
2.7 DAQ sequence number 2: multi-event test sequence 13
2.8 DAQ sequence number 3: single-event test sequence 14

3.1 Format of the ANAPIL test log file for a given DAC value. The value with an * are from ADCs on the MPT board. . . . 19

4.1 DIGITAL PILOT automated test's PHASE LOG format . . . 21
4.2 DIGITAL PILOT automated test's EROR CONTROL LOG format 21
4.3 DIGITAL PILOT automated test's EROR SLOT LOG format 22

6.1 USB header format (to be repeted twice) 24

List of Figures

1.1 The MPT board. On the bottom the connector for the MCC board. 7

Chapter 1

Introduction

The MCM's Production Test system (MPT system) consists mainly of 2 boards:

The MCC or MCM Carrier Card , a board whose sole pourpouse is to be a stable and rigid support for the MCM under test and its fragile optical apparatus

The MPT or MCM Production's Test board on which sits all the logic and the active components used for the test and qualification of the MCM

The 2 boards are connected via a 100 pin connector and 3 optical links. The power supply for the MCC comes from the MPT. The developed test for the production are authomatically performed and the results are stored in log files.

The Log files are stored in the memory of the MPT and in some of its internal registers and are accessible via USB or VME using the support of the Rudolf board [1].

[1]The test board developed for the prototyping of the full read out chain of the ALICE's SPD

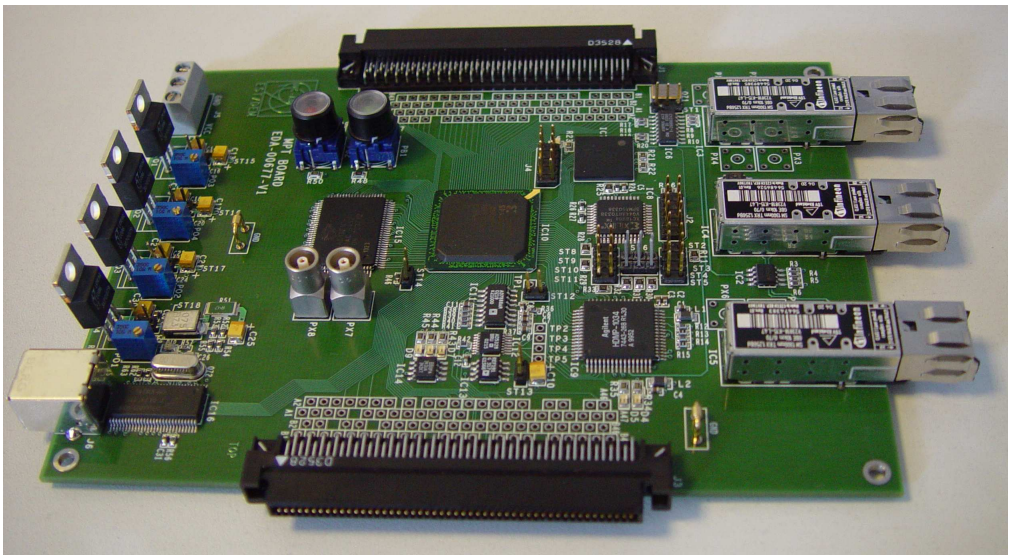

Figure 1.1: The MPT board. On the bottom the connector for the MCC board.

Chapter 2

MPT registers

The MPT contains several registers for its control and status' monitor. Via these registers manual and fully authomated test can be performed. In table 2.1 are reported all the registers and their address.

2.1 STATUS register

This is a read only register reporting the status of the test board. The meaning of its bits is reported in table 2.2

2.2 COMMAND register

This is the main register of the FPGA. The MPT is controlled via this register. The meaning of its bits is reported in table 2.3. The 8 MSBs of this register define the command we want to execute, the rest are command specific bits. After the execution of each command the MSB of this register is set to 1. The availeable commands are listed with their code in the table 2.4.

2.2.1 READ MEMORY command

This command starts the transfer of the content of the memory. Being the memory a 32 bit one each double word is transferred in two times : before bits 31–16 and then 15–0.

NAME	ADDRESS	R/W
STATUS	0	R
COMMAND	1	R/W
TDI [191:160]	9	R/W
TDI [159:128]	10	R/W
TDI [127:96]	11	R/W
TDI [95:64]	12	R/W
TDI [63:32]	13	R/W
TDI [31:0]	14	R/W
TDO [191:160]	15	R
TDO [159:128]	16	R
TDO [127:96]	17	R
TDO [95:64]	18	R
TDO [63:32]	19	R
TDO [31:0]	20	R
DAQ COMMAND	21	R/W
JTAG VALIDATION LOG	23	R
ANAPIL STATISTICS	24	R/W
ANAPIL STEP VALUE	25	R/W
ANAPIL STEP'S NUMBER	26	R/W
ANAPIL WAIT AFTER DAC	27	R/W
ANAPIL WAIT BETWEEN ADC	28	R/W
ANAPIL MCM VDD	29	R/W
JTAG MEM	126	W

Table 2.1: MPT's Registers

2.2.2 START JTAG command

This command starts the a JTAG access of the MCM.
Bit 12 of COMMAND register is used to select which kind of access to perform: if set to one the JTAG controller will perform an IR access and a DR one in the other case. The number of shifts to perform has to be written decremented by 1 in bits 11–0 of COMMAND register.
The TDI sequence can be taken either from the JTAG memory or from the TDI register. From which memory to take the stream, and where to store the result, is selected by bit 13 of the command register: a 1 in that bit selects the JTAG mem while a 0 the TDO register.
When the sequence for TDI is taken from TDI register the maximum number of shifts allowed is 192. The first bit to be shifted out is TDI[0]. The read

BITS	MEANING
31–26	Unassigned (0)
25	Digital Pilot test IDLE
24	JTAG chain validation IDLE
23	Analog Pilot test IDLE
22	Synchronization procedure performed
21	DACs ready to be programmed
20	Programming a DAC
19	ADCs ready to perform conversion
18	Reading the ADCs
17	DAQ block running
16	JTAG running
15–12	G-Link synch's delay value, see 4
11–8	Serial synch's delay value, see 4
7–0	Copy of the JTAG validation log, see 2.2.9

Table 2.2: STATUS register's bit definition

BITS	MEANING
31–24	Command
24–0	Command specific configuration's bits

Table 2.3: COMMAND register's bit definition

sequance is stored in TDO register. The bits are shifted in from TDO[191]. At the end of each access the content of TDO register or of the TDO memory is transferred starting by the less significant word (16 bit).

2.2.3 READ ADC command

This command starts the ADC read out. At the end each of the 8 read values is transferred with prefixed (on bits 15–12) its ordering number.

2.2.4 PROGRAM DAC command

This command starts the programming of one of the 8 DACs on the board. The value at which the DAC will be programmed to must is specified in bits 11–0 of COMMAND register, while the DAC to be programmed is selected using bits 14–12.

COMMAND	CODE
Read Memory	1
Start JTAG	2
Read ADC	3
Program DAC	4
Start DAQ	5
Read Register	6
Initialize Memory	7
Synchronize	8
Validate JTAG Chain	9
Anapil scan	a
Digital Pilot Test	b
Invert command polarity	c
ANAPIL ADC scan on	d
ANAPIL ADC scan off	e
USB mode 1	10
USB mode 2	11
JTAG internal controller	12
JTAG external controller	13

Table 2.4: Availeable commands

2.2.5 START DAQ command

This command will start a series of DAQ sequences for the MCM. The number of sequences to be performed is specified in bits 7–0 of the command register, while the sequence type is specified in bits 11–8. There are 4 preprogrammed sequences and a programmable one (sequences number four). A detailed description of the preprogrammed sequences is reported in tables 2.5 to 2.8.

COMMAND	time(100ns)
wait	201
L1	1
wait	6
L2Y	4
wait	2703

Table 2.5: DAQ sequence number 0: synch sequence

COMMAND	time(100ns)
wait	207
reset_bob	3
wait	120
reset_global	3
wait	3
TRST_pixel	3
wait	3
TRST_ON	3
wait	3
TRST_OFF	3
wait	3
reset_gol	3
wait	4
reset_pixel	3

Table 2.6: DAQ sequence number 1: reset sequence

2.2.6 READ REGISTER command

This command starts the transfer of the value of the register whose address is specified in bits 6–0.

2.2.7 INITIALIZE MEMORY command

This command start the initialization of the MPT memory with a predefined pattern. This pattern starts with the hexadecimal value deadbeef and continues with incrementing values starting from 00020001, 00040002.....

2.2.8 SYNCHRONIZE command

This will start a synchronization process: a DAQ sequence of type 0 will be sent and used to calculate the delay to apply to the serial command and to the glink control to and from the simulated pilot to have it synchronized with the one under test. The result of this operation is reported in the STATUS register, but it will not be authomatically transmitted at the end of the operation.

COMMAND	time(100ns)
wait	201
test_pulse	4
wait	11
L1	1
wait	6
L1	1
L2N	4
wait	6
L1	1
wait	6
L2Y	4
wait	6
L1	1
wait	6
L1	1
wait	6
L2N	4
wait	6
L2Y	4
wait	6
L2N	4
wait	6006

Table 2.7: DAQ sequence number 2: multi-event test sequence

2.2.9 VALIDATE JTAG CHAIN command

This command start a procedure of validation of the MCM's JTAG chain in all the possible configurations. The log file of the test is transmitted as a 8 bit register at the end of the test itself. The position of each bit in this register reflects the 3 bits configuration in the digital pilot as they where during the test. As an example a 1 in position 0 means that the validation succeded in "TDO_PIX9 & SKIP_NONE" mode. The log of the test is also present in STATUS register at bits 7–0.

2.2.10 ANAPIL SCAN command

This command initiates the scan for the characterization of the ANAPIL DACs and ADCs. The result of such a scan is saved in memory and is

COMMAND	time(100ns)
wait	201
test_pulse	6
wait	21
L1	1
wait	12
L2N	4
wait	6
L1	1
wait	6
L2Y	4
wait	2703

Table 2.8: DAQ sequence number 3: single-event test sequence

not authomatically sent. For a description of the operation and of the data format in the memory see 3

2.2.11 DIGITAL PILOT TEST command

This command initiates the authomated test of the DIGITAL PILOT. The results of the test are stored in a log file in the memory of the MPT, and are not authomatically transferred at the end of the test. For a detailed description of the test and of the log file format see 4

2.2.12 INVERT COMMAND POLARITY command

This command changes the polarity of the serial stream for the MCM. At the same time it also changes the polarity of the push button led that works at this moment also as command polarity indicator.

2.2.13 ANAPIL ADC SCAN ON command

This command will reactivate the scan of the ANALOG PILOT ADC during the ANAPIL test. By default it is active.

2.2.14 ANAPIL ADC SCAN OFF command

This command will deactivate the scan of the ANALOG PILOT ADC during the ANAPIL test. The readout of the ADC is done, but the conversion is

not. By default it is active.

2.2.15 USB mode 1 command

This command will set the mode of the USB interface to USB1. By default is in mode USB2.

2.2.16 USB mode 2 command

This command will set the mode of the USB interface to USB2. By default is in mode USB2.

2.2.17 JTAG internal interface command

This command will enable the internal JTAG interface disabling the external spare connector pins.

2.2.18 JTAG external interface command

This command will disable the internal JTAG interface and activate the MPT spare connector for the reception of JTAG streams.

2.3 TDI and TDO registers

Those are the registers where it is stored the incoming and the outcoming data JTAG data for the JTAG controller.

2.4 DAQ COMMAND register

This is a register used to map the memory in which it is possible to store a custom DAQ sequence.
The memory is 16 bit whide and 256 word deep. The address to which write is specified by bits 31–24. The memory is read sequentially: each read access makes the memory pointer increase. The memory pointer for the read is resetted at each write access.The read location's address is anyway reported as crosscheck in bits 31–24.

2.5 JTAG VALIDATION LOG register

This is the register where the result of the JTAG validation test is stored. To have information about the format see 2.2.9

2.6 ANAPIL STATISTICS

During the ANAPIL test the ADCs are read several times for each DAC value and the readout values accumulated. The number of readout to perform for each DAC value is stored decremented by 1 in this register. Only bits 5–0 are valid, making the maximum number of readout per value equal to 64.

2.7 ANAPIL STEP VALUE

This is the amplitude of which the MPT DACs are incremented at each step during the ANAPIL scan.

2.8 ANAPIL STEP'S NUMBER

This register fixes the number of values we want to scan (max 4096 setting it at 4095).

2.9 ANAPIL WAIT AFTER DAC

This is the time in 25ns units we want to wait after a change in a DAC to be sure it is at a stable value.

2.10 ANAPIL WAIT BETWEEN ADC

This is the time in 25ns units we want to wait between successive ADC readout to avoid influencing the result.

2.11 ANAPIL MCM VDD

The VDD/2 value on the MCM as read from the ANAPIL is not reported in the ANAPIL test log because it is not possible to make a scan out of it. Nevertheless this value is read and accumulated into this register during the scan.

2.12 JTAG mem

This is the address of a 4Kb memory used to store the TDI stream for the JTAG controller as an alternative to the TDI register in case of need of streams longer than 192bits.

Chapter 3

ANAPIL's DACs and ADCs scan and characterization

To test the ANAPIL the outputs of its DACs are read by 12 bit ADCs sitting on the MPT board, and the inputs of its ADCs are stimulated by 12 bit DACs sitting on the MPT board. The LSB of a MPT's DAC or ADCs is set to 0.5V using an external precision reference.
The test consists of the repetition "ANAPIL STAT'S NUMBER" +1 times of the following steps:

1. loading the all the MPT DACs with the same 12bit value ("DAC VALUE") and the ANAPIL DACs with the 8MSB of it
2. wait for the DACs output to stabilize "ANAPIL WAIT AFTER DAC"*25ns
3. repeat "ANAPIL STATISTICS"+1 times the following steps
   - read the ANAPIL DACs with the MPT ADCs
   - read the ANAPIL and MPT DACs with the ANAPIL ADCs (the ANAPIL ADCs conversion can be disabled and the read value is always the same in this case)
   - accumulate in apposite registers the read values
4. increment "DAC VALUE", initialized at the beginning of the test to 0, of "ANAPIL STEP VALUE"

At the end of each of those cycles a log file is written in the MPT memory. The log file format for a given dac value reported in table 3.1. At the end of the test there are "ANAPIL STAT'S NUMBER" +1 of those log files stored in memory starting from address 0.

memory cell	bits 31..16	bits 15..0
n	4'b0, DAC value	DAC_ref_hi *
n+1	DAC_ref_mid *	Test_Pulse_hi *
n+2	Test_Pulse_low *	GTL_ref *
n+3	T1	DAC_ref_hi
n+4	DAC_ref_mid	GTL_ref
n+5	Test_Pulse_hi	Test_Pulse_low
n+6	Pixel_VDD/2	SENSE_V
n+7	SENSE_I	SENSE_I 2

Table 3.1: Format of the ANAPIL test log file for a given DAC value. The value with an * are from ADCs on the MPT board.

It is important to note that the accumulators for the MPT's ADCs are 18bits wide to cover the maximum possible statistics requirement, but in the log file only the 16MSB are reported. It is also worted to mention that for the ANAPIL's DACs only the 8MSB of the "DAC value" field in the log file are valid.

Chapter 4

DIGITAL PILOT automated test

With this test we want to check the DIGITAL PILOT functionality in all its possible configuration. To do this the pilot response to reset and acquisition sequences is compared with a pilot model instantiated inside the FPGA. The test is performed for 4 different phases between the clock and the serial control stream (0, 90 180, 270). It is also checked the capability of the DIGITAL PILOT to synchronize its internal 10MHz clk on the idle command stream.
The first thing we do is to check if the pilot can understand commands with a given phase respect to the clock. To do this a full reset sequence is sent to the MCM and we check that all the reset signals are properely sent to all the chips. At this time we also check for the GOL relock time after a reset. If all the reset signals are properely received and the the GOL relocked the test proceed with the synchronization of the emulated pilot in the FPGA with the one under test. At the end of this procedure, or at the first error encountered, a PHASE LOG (table 4.1)is written.
If bit 6–0 of the PHASE LOG are not all at 1 it means that at leat one of the test failed and the test repeat the reset and synchronization test with the next clock-serial stream phase. In case of success (bit 6–0 all at 1) the DIGITAL PILOT is tested for 4 different idle command phase (steps of 25ns) and for 16 different combinations of its intenal configurations registers. The 16 configurations are studied to cover all possible configuration of each register inside the pilot. During the test the pilot is stimulated with a sequence of command studied to cause all the possible answers from it. Possible errors in the pixel and gol controls and in the communication are flagged in 5 32 bit words of log stored in memory. The first one reports the errors in the control signals, the remaining 4 in the comunication, one for each slot starting from

BIT	MEANING
31–28	Unassigned (0)
27–24	GLINK Synch delay
23–20	SERIAL Synch delay
19–8	GOL relock time (6.4us)
7	Unassigned (0)
6	Synchronization done
5	Synchronization started
4	GOL relocked
3	GOL resetted
2	Reset GOL signal sent
1	TRST Pixel signal sent
0	Reset Pixel signal sent

Table 4.1: DIGITAL PILOT automated test's PHASE LOG format

slot0. The format of the error control log is reported in table 4.2, while the format of the other 4 is the same and can be seen in table 4.3.

BIT	MEANING
31–24	Number of errors
23–21	Unassigned (0)
20	CAV error
19	DAV error
18	Glink not ready
17	Reset GOL signal error
16–7	CE9–CE0 error
6	Test Pulse error
5	Strobe error
4	ClEv error
3	NEvR error
2	ShRst error
1	Reset Pixel error
0	TRST Pixel error

Table 4.2: DIGITAL PILOT automated test's EROR CONTROL LOG format

At the end of the test the LOG file in memory will be of a minimum of 4 32 bit words in case of failure of all the phase check to a maximum of

BIT	MEANING
31–24	Number of errors
23–20	Unassigned (0)
19	HP_rx not ready
18	HP_rx error flag
17	CAV error
16	DAV error
15–0	LINKBUS error

Table 4.3: DIGITAL PILOT automated test's EROR SLOT LOG format

4*(1+19*5)=384 32 bit words in case of complete test. The log file is written in the memory starting at address 8000(in hexadecimal) to avoid overlapping with the ANAPIL characterization log.

Chapter 5

The VME interface to the MPT

To write a register in the MPT board via the VME you must write the value you want to write at address 32'h02000008 and than the address of the register you want to access at address 32'h02000004. This will initiates the serial transfer from Rudolf to the MPT. The outcome of each command, if any, is written sequentially in the 16 LSB of the memory starting always from position 0 of the Rudolf memory. After each command the memory should be read because the next command output will overwrite it. To read an MPT's register or its memory it is necesaary to use the apposite commands of the MPT(see 2.2). The 32bit words are transmitted MSB first.

Chapter 6

The USB interface to the MPT

The access to the MPT via USB is made in blocks of 32bit words. It is sufficient to send the data we want to transfer prefixed by a header specifying how many 32 bit words are in the packet and the address to which write the first of those words. The header is repeated twice to avoid unwanted memory access. The exact structure of the header is reported in table 6.1. Each 32bit word is trasferred as 2x16bit, LSB first. It is important to note that if the

BIT	MEANING
15	1'b0
14–8	start address
7–0	packet lenght

Table 6.1: USB header format (to be repeted twice)

start address is 126 it is assumed that all the packet has to be written to this address, while in all the other case the address is authomatically incremented. Address 127 is not valid.